A Survey of Current Android Attacks

Timothy Vidas
ECE/CyLab
Carnegie Mellon University

Daniel Votipka
INI/CyLab
Carnegie Mellon University

Nicolas Christin
INI/CyLab
Carnegie Mellon University

Outline

- Introduction
- Android Security Model
- Security Model Analysis
- Attack Taxonomy
- Possible Mitigations

Introduction

- Adoption of smartphones has increased
  - Nearly 50% of cell phones sold are smartphones
- Ubiquity of the market
- Devices are faster, more connected, and always on
- Smartphones hold sensitive user information
  - Banking Information
  - Current Location

Introduction

- Major smartphones use a managed model of service
  – Google and Carrier control administration
- Closer to corporate-managed devices than personal machines
  – Managers control security instead of user

Android Security Model

- Android was designed with security in mind
- Sandboxes applications
  - Apps can only access their own data
  - Apps divided through privilege separation
- System resources accessed through permissions

Permission Model

- Applications require explicit permission from users at install time
- Intended to prevent unwanted access to user data

Android Market

- Un-moderated Market
- Users flag applications
- Remote Kill

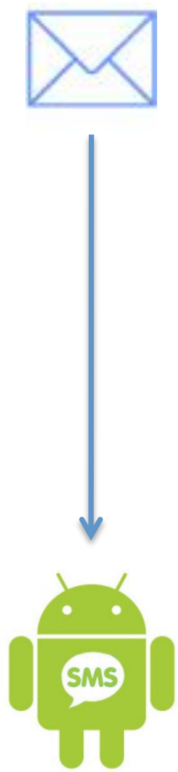
Security Model Analysis

Android Permissions

- Hard for user to understand
  - ACCESS_SURFACE_FLICKER and BIND_APPWIDGET
- Permissions can be too general
- Unexpected consequences

RECEIVE_SMS

- Malicious apps can take advantage of permissions



RECEIVE_SMS

- Malicious apps can take advantage of permissions

Patch Cycle

Not Android Specific

Android Specific

Vulnerability Discovered

Component Patch Available

Manufacturer Releases Patch

User Applies Patch

A → B → C → D → E → F → G

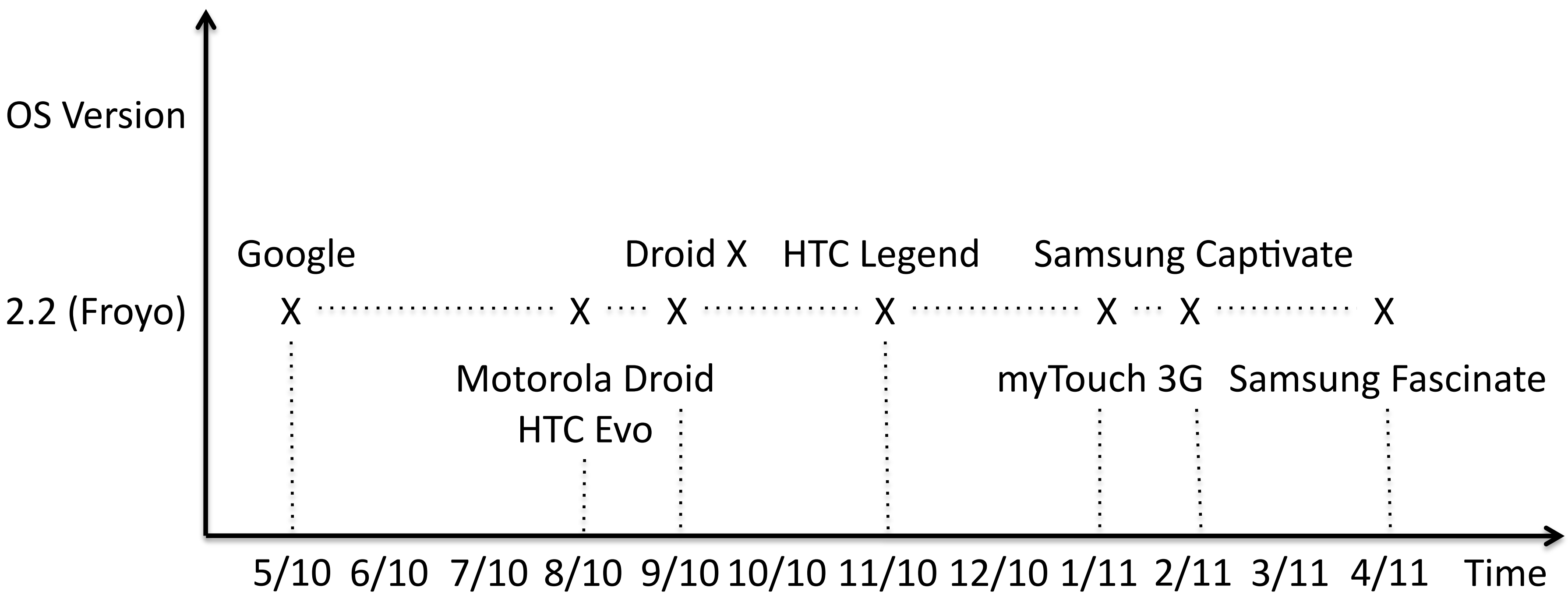
Vulnerability Disclosed

Google Releases Patch

Carrier Releases Patch

Exploit Window

Patch Cycle


OS Version
2.2 (Froyo)
Google
Droid X
HTC Legend
Samsung Captivate
X X X X X X X
Motorola Droid
HTC Evo
myTouch 3G
Samsung Fascinate
5/10 6/10 7/10 8/10 9/10 10/10 11/10 12/10 1/11 2/11 3/11 4/11
Time

Patch Cycle

- Attacks stay viable longer
- Attackers can build exploits through analysis of early updating devices

Trusted USB Connection

- Android Debug Bridge (ADB) developer tool
  - Gives access to interactive shell
  - Allows developer to push applications directly to the device
- ADB doesn't require authentication
- Attacker can use ADB to bypass Android Market

Recovery Mode

- Circumvents standard boot partition
- Allows user to recover from “bricked” phone
- Attacker can use to install malicious image

Uniform Privilege Separation

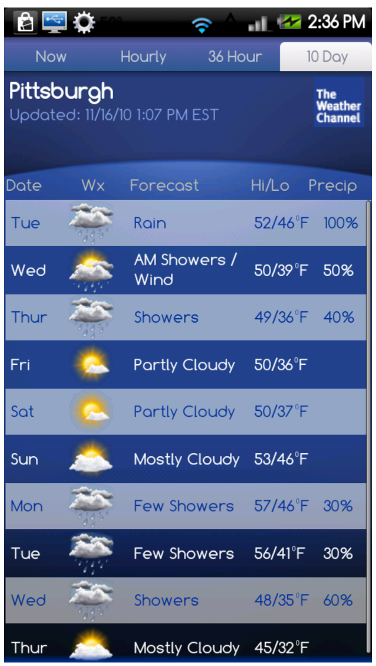
- Security tools typically require root access
- Android restricts all apps the same

Attack Classes

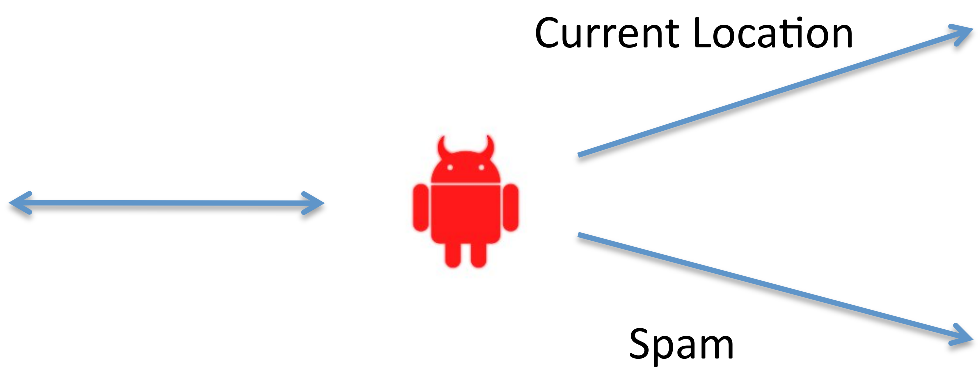
Unprivileged Access

- Operate within the permission system

Foreground



Background



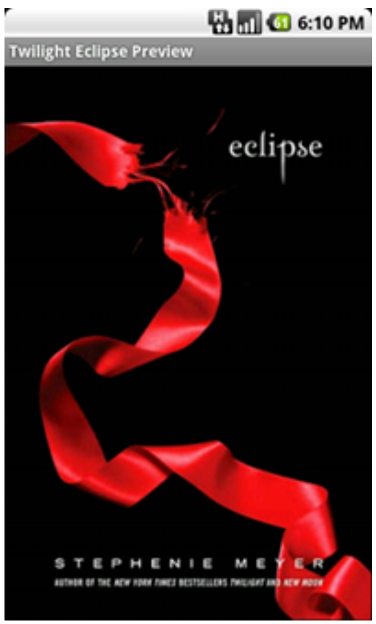
Unprivileged Access

- pjapps (aka Andr)
- Geinimi
- FakePlayer
- DroidDream
- Bgserv
- Ggtracker
- Hipposms
- YZHCSMS
- HTCFakepatch
- GoldDream
- DroidKungFu
- DroidKungFu2
- jSMSHider
- BaseBridge
- DroidDreamLight
- EndOfDays
- Zsone
- zitmo (aka zues)

Remote Exploitation

- RootStrap Attack*

Foreground



Background

Privilege
Escalation Exploit

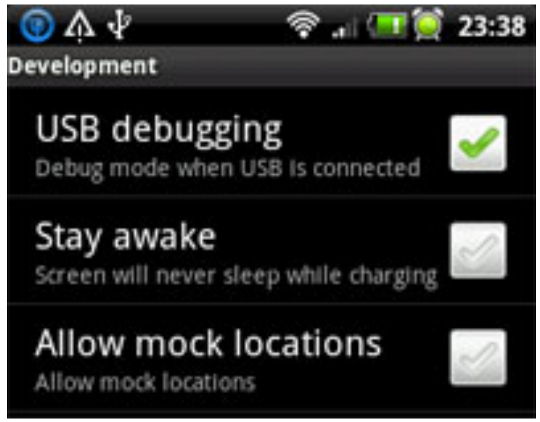
*Jon Oberheide, In SummerCon, 2010

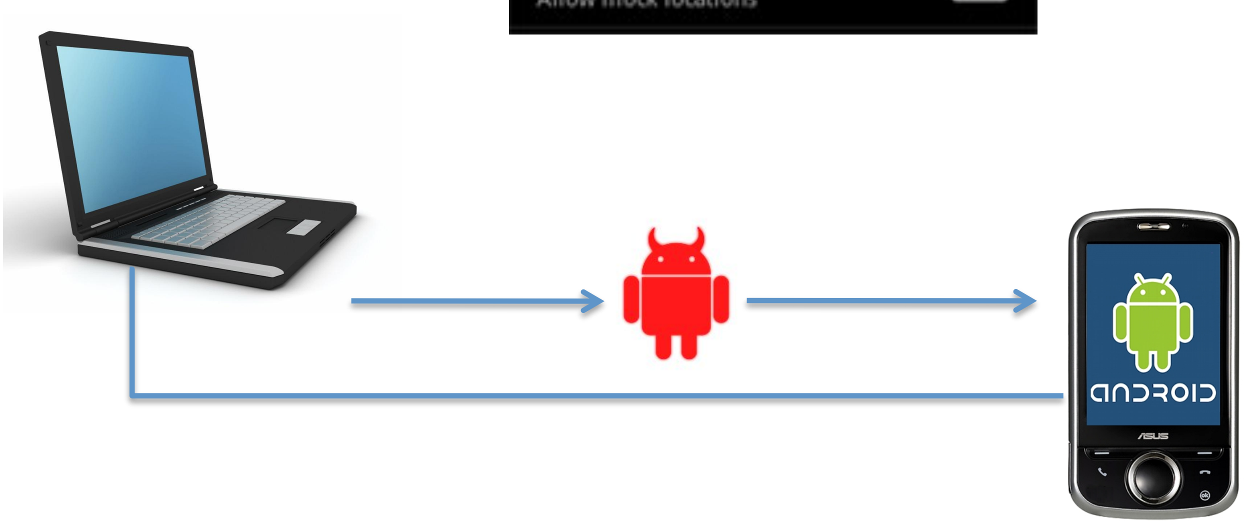
Remote Exploitation

- Legitimate Rooting Applications
- WebKit Vulnerabilities*

*CANVAS 6.65, www.immunityinc.com/

Physical Access With ADB Enabled





Physical Access with ADB Enabled

- Super One-Click desktop application
- Minimal device modification
  – Hard for non-rooted devices to detect

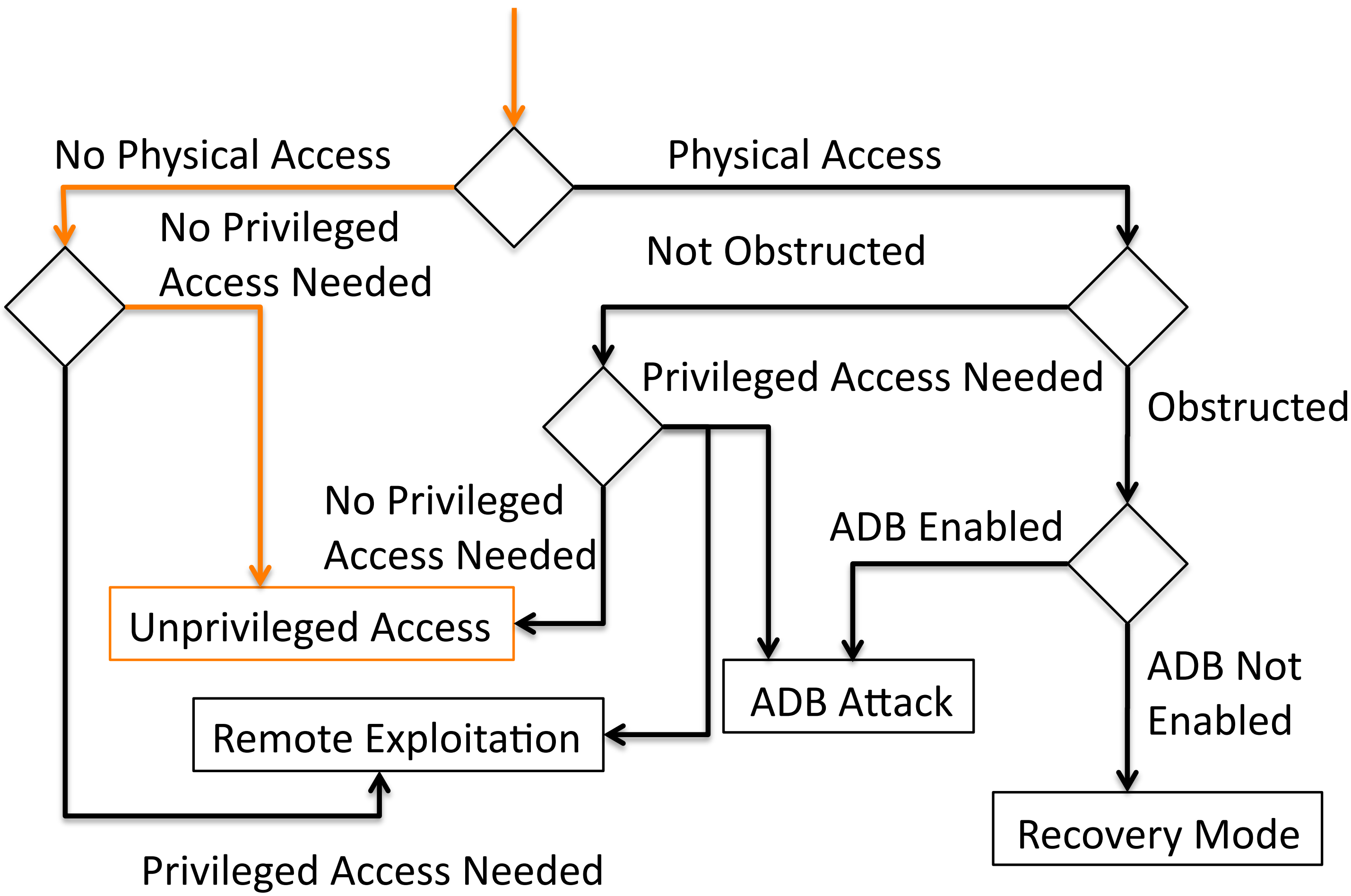
Recovery Mode

- Create a custom recovery image to gain root access
- Bypass authentication by using the recovery mode

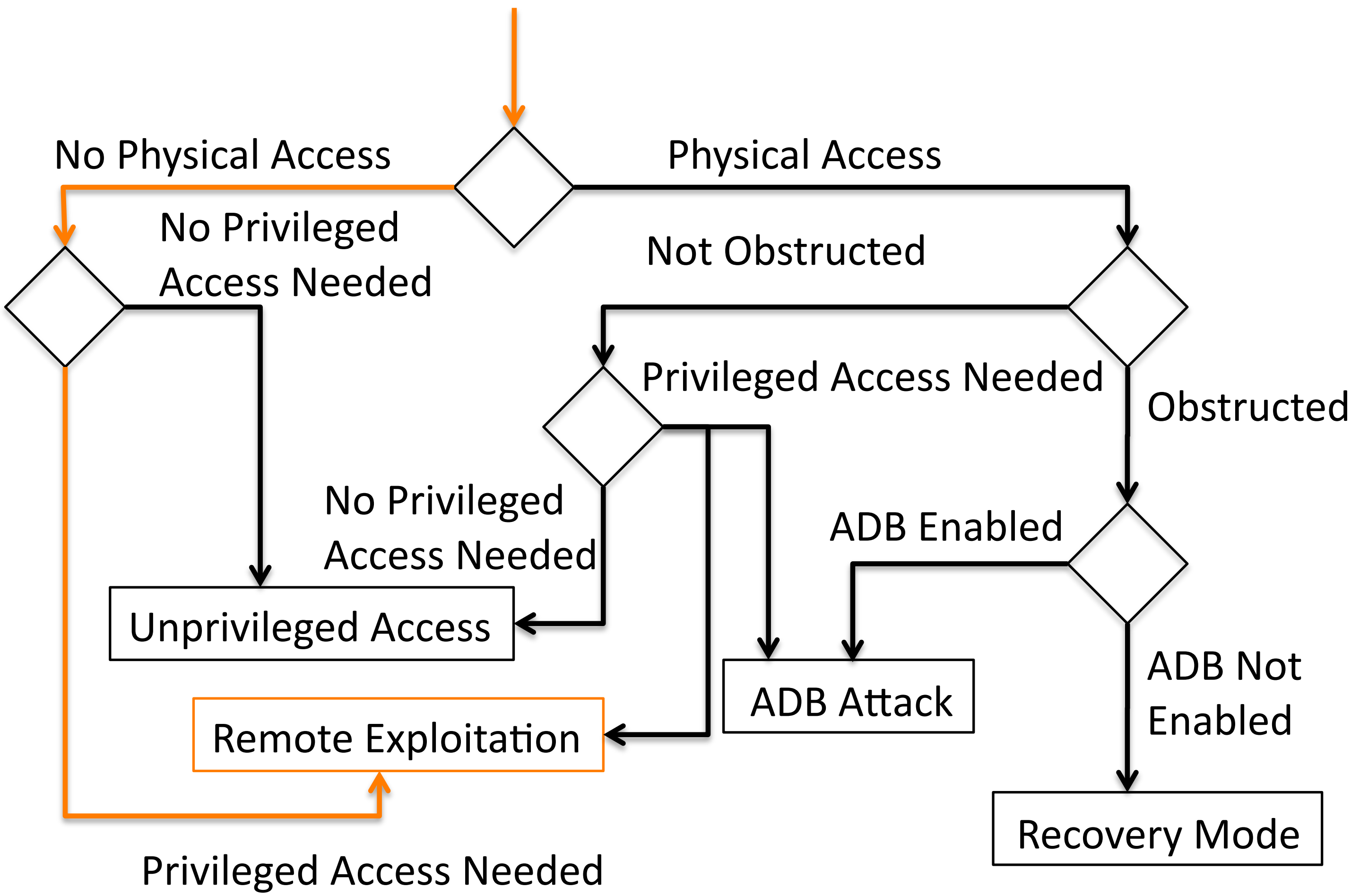
Recovery Mode

- Does not rely on a kernel exploit
- Large footprint, but simple to cover your tracks

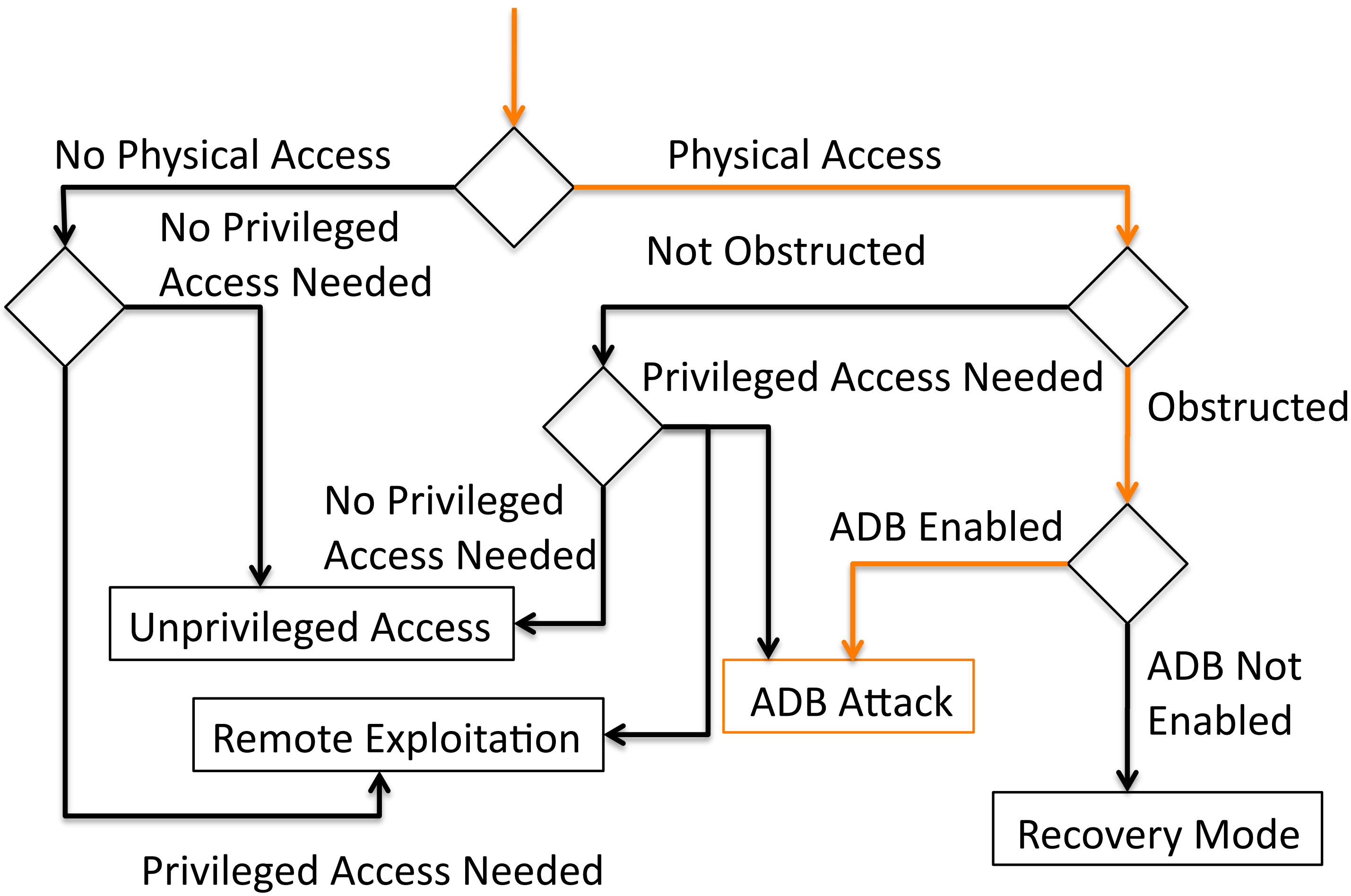
Attack Chart



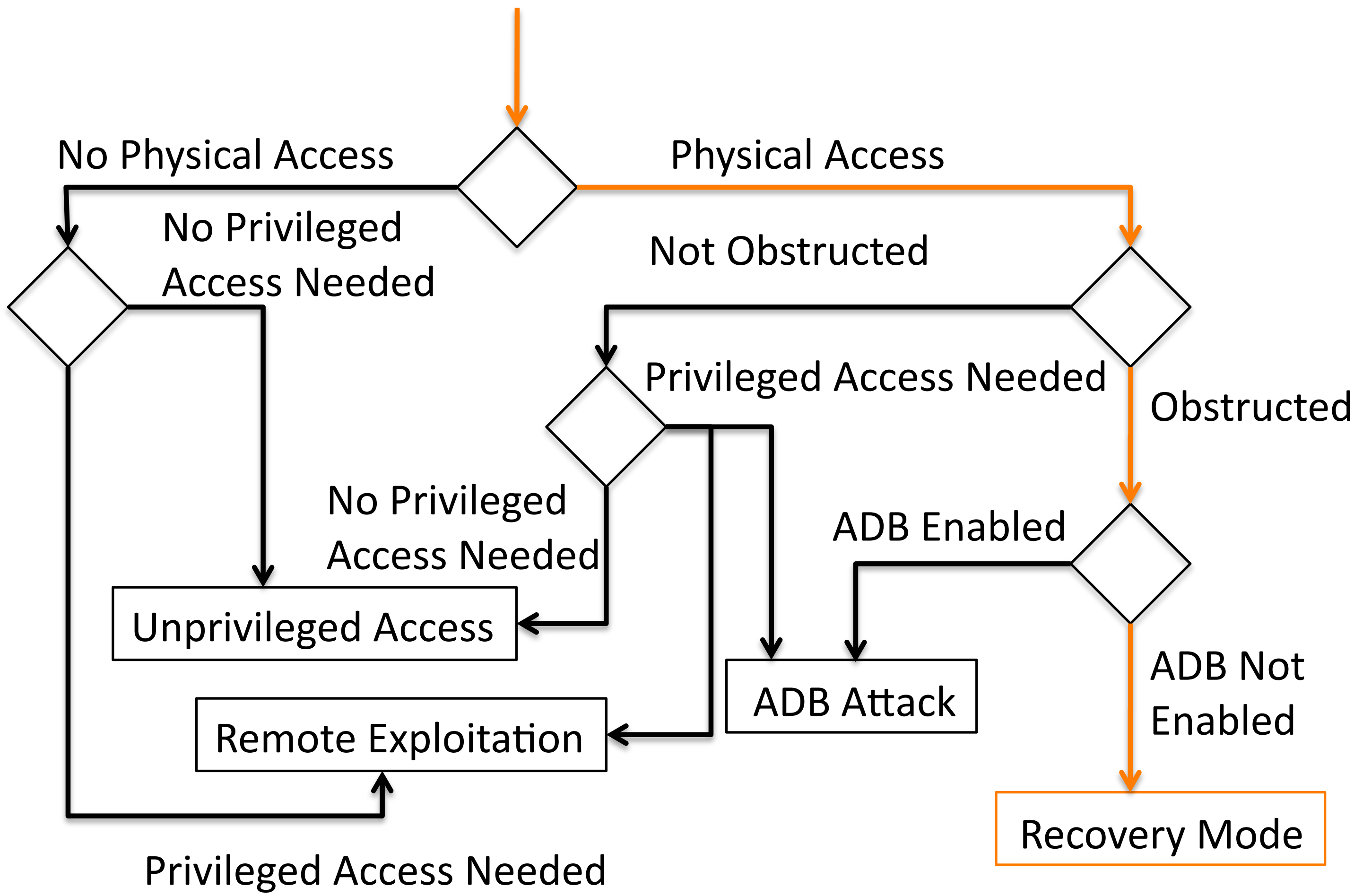
Attack Chart



Attack Chart



Attack Chart



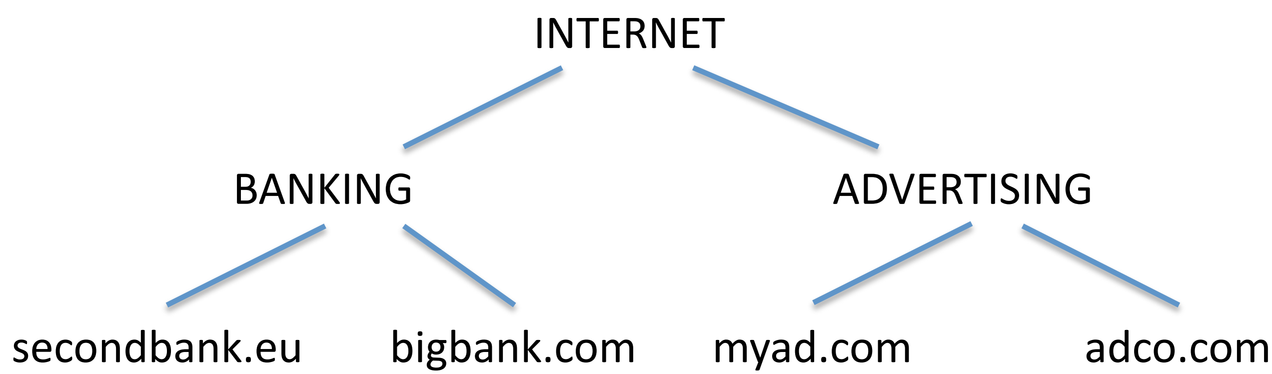
Possible Mitigations

Reduce Patch Cycle Length

- Google produces patches relatively quickly
- Separate manufacturer modifications from core of Android

Adjusted Permission Model

- Hierarchical permissions

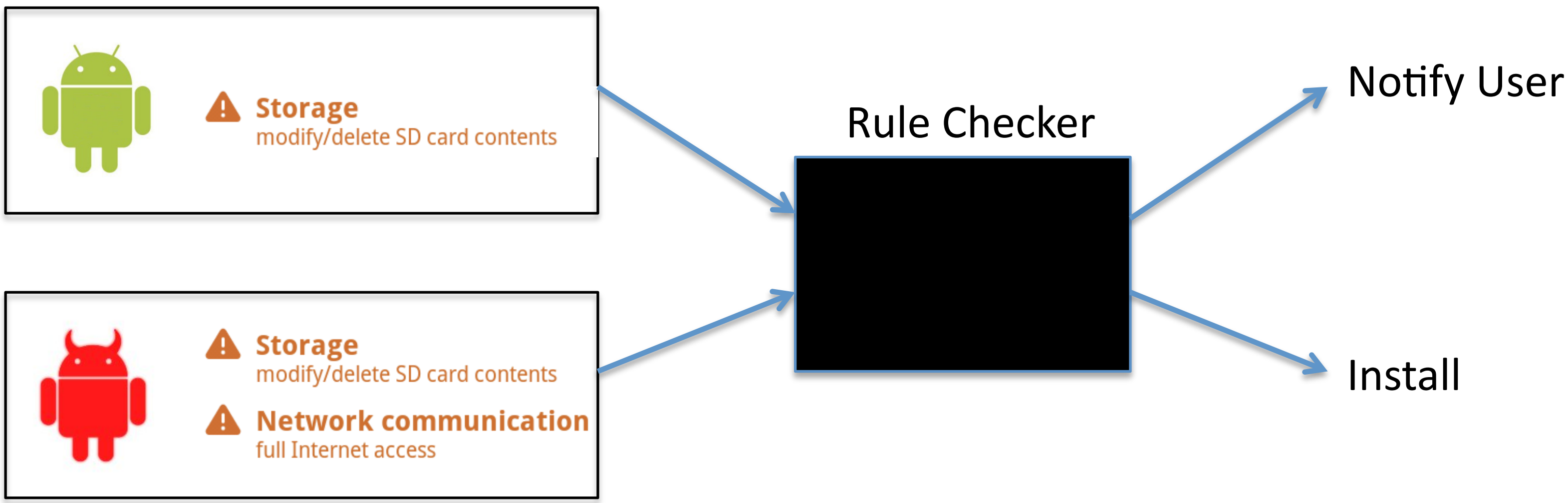


- Would require a re-structuring of the permission model

Barrera et al, In *ACM Conference on Computer and Communications Security*, 2010

Adjusted Permission Model

- Check requested rules for possible malicious combinations



Enck et al, In *ACM Conference on Computer and Communications Security*, 2009

Adjusted Application Privilege

- Two classes of applications:
  1. Standard Applications – Same privileges and rights as current apps.
  2. Privileged Applications – Root privileges, but must under-go a moderated review before publication
- Allows the user to trust some apps that watch the others

Leverage Existing Technology

- Port operating system concepts such as SELinux, ASLR, or Firewalls to the Android kernel.
- Implement something similar to TaintDroid* framework to give real-time information on permission usage
- Processing costs must be considered

*Enck et al, In *Proceedings of OSDI*, 2010

Additional Authentication

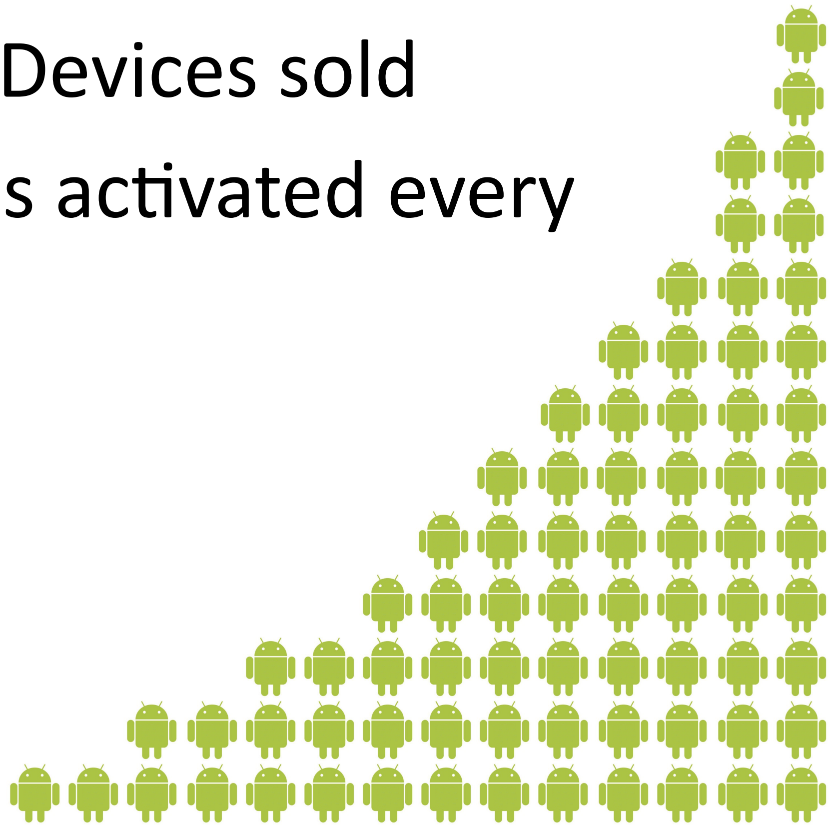
- Require user credentials to install an Application
  - Currently done on the iPhone
- Require authentication for ADB tool
  - Removes backdoor around locking mechanism

Trusted Platform Module

- Provides ground truth for device security
- Mitigates recovery image attack

Conclusion

- 100M Android Devices sold
- 500,000 devices activated every day



Conclusion

- 100M Android Devices sold
- 500,000 devices activated every day